On the Verification of Deep Reinforcement Learning Solution for Intelligent Operation of Distribution Grids

Mohammad Mehdi Hosseini, Masood Parvania
Department of Electrical and Computer Engineering, The University of Utah, Salt Lake City, UT 84112
E-mails: mehdi.hosseini@utah.edu, masood.parvania@utah.edu

Abstract

Capabilities of deep reinforcement learning (DRL) in obtaining fast decision policies in high dimensional and stochastic environments have led to its extensive use in operational research, including the operation of distribution grids with high penetration of distributed energy resources (DER). However, the feasibility and robustness of DRL solutions are not guaranteed for the system operator, and hence, those solutions may be of limited practical value. This paper proposes an analytical method to find feasibility ellipsoids that represent the range of multi-dimensional system states in which the DRL solution is guaranteed to be feasible. Empirical studies and stochastic sampling determine the ratio of the discovered to the actual feasible space as a function of the sample size. In addition, the performance of logarithmic, linear, and exponential penalization of infeasibility during the DRL training are studied and compared in order to reduce the number of infeasible solutions.

1. Introduction

Deep reinforcement learning (DRL), the combination of deep neural network (NN) with any of reinforcement learning algorithms, has proven effective in finding near-optimal action policies in partially observable problems, where the state transition probabilities are unknown or hard to obtain. For instance, DRL has been trained to play various video games by numerous interaction with frames of the game environment, and has achieved human-level control [1]. Mastering the Korean game Go, whose state space has a massive size and had been a long lasting problem for computer scientists, was also achieved by combining DRL with Monte Carlo Decision Tree [2]. Fast response time of DRL-trained agents, as well as its capabilities in unsupervised search of high-dimensional and highly uncertain problems, have led researchers to suggest it as an alternative solution for operational research [3, 4].

Operation problems typically require an optimal solution over a time horizon and quite often are subject to uncertainty sources. Stochastic optimization is the common method to solve operation problems under uncertainty, but it can be too slow for real-time applications, and struggle when dealing with roughly-modeled uncertainties. Accordingly, DRL has found its place in power systems operations to replace classical mixed-integer stochastic optimization models. Researchers have adopted DRL, either with continuous or discrete action space, in operating DER, storage units, volt-var control devices, generating retail prices, and dynamic market matching [5–9]. Real-time operation of DER has been a popular field for using DRL, where simple actor-critic models [10] or Asynchronous Advantage Actor Critic (A3C) algorithm [11] are used to generate control signals that minimize the long-term operation costs. Discrete Deep Q-Network (DQN) is also used in [12] to select optimal long-term dispatch level of storage devices. Scalablity issue of controlling multiple DER units with DRL is investigated in [13] using a Deep Deterministic Policy Gradient (DDPG) with sequential decision-making.

A major concern in using DRL for critical operations is the lack of guarantee for a feasible solution that satisfies all problem constraints. In DRL, infeasiblity is typically avoided by penalizing impossible solutions during training or clipping the solution within its direct limits, however, none of them ensure that all constraints are met when a deep NN is used as a black box. To overcome this hurdle, researchers have exploited the piece-wise linear structure of NN models to provide upper and lower feasiblity bounds using Satisfiablity Modulo Theory [14, 15]. These methods are based on linear approximation or exact representation (in the case of ReLU) of the activation function in the NN structure, and obtaining a closed-form formulation of NN that can be checked against problem constraints. This approach is also adopted in power system research for finding adversary examples [16] and worse case guarantee [17] when NN is used as the operational

URI: https://hdl.handle.net/10125/79761
978-0-9981331-5-7
(CC BY-NC-ND 4.0)

decision-maker. The verification problem is even more troublesome when an NN is trained within unsupervised frameworks such as DRL. In these models, using adversary examples in the training is not as effective in correcting their behavior, and it is critical for the operator to know the feasibility space before using DRL for decision-making. Also, the impact of infeasibility penalization on reducing the number of impossible solutions has not been investigated.

This paper first formulates the operation of DER units in a distribution system as a Markov Decision Process (MDP), based on which a DRL model is developed and trained to operate the distribution system. Then, the structure of the trained NN within the DRL is reformulated by a linear binary model, which is combined with a quadratically constrained (QC) formulation of distribution grid constraints, allowing us to define an optimization model to find the largest feasible sphere around each feasible sample. Robustness margins are also defined for each feasibility sphere, to account for potential noise or perturbations in the observed system state. The feasible space is explored by finding feasible spheres around numerous sample points, and aggregating all spheres in a feasibility set. The obtained feasible set is then used when applying DRL on real systems, to separate system states for which the DRL solution is potentially infeasible. Empirical studies are performed to find the required number of samples for effective discovery of feasible space. Further, the impact of logarithmic, linear, quadratic and cubic penalization of infeasibility during DRL training is measured through extensive studies on three test systems with various sizes, and the results are compared against each other.

The rest of the paper is organized as follows: In Section 2, the power system operation is presented as an MDP, and a DRL framework is defined for solving the problem. Section 3 presents the method to find the feasibility set by aggregating the largest feasible spheres around sample points. The method is then implemented on three test distribution networks in Section 4 where the feasibility set is used to identify system states in which the DRL agent produce infeasible results. Finally, the paper is concluded by Section 5.

2. Distribution Grid Operation using DRL

In order to find the optimal operating point of distributed generators (DG) and energy storage (ES) units within a distribution grid, which is exposed to uncertainty, the following cost minimization problem is solved for N stochastic scenarios to find the least expected operation costs over time horizon T:

$$\textbf{P1:} \min_{\mathbf{u}\in\mathbf{U}} \sum_{t\in T}\sum_{n\in N} C_{n,t}(\mathbf{s},\mathbf{u}), \tag{1}$$

$$\text{s.t. } \textit{Constraint (37)-(45) in Appendix A} \tag{2}$$

where $C(\mathbf{s},\mathbf{u})$ is the operation cost as a function of control variables $\mathbf{u}$ and system states $\mathbf{s}$, and the problem is subject to distribution grid constraints, which are formulated in quadratically constrained form in Appendix A.

DRL solves the distribution grid operation problem by training a NN that takes system state as the input and gives a set of control variables with the highest long-term reward. This section designs a DRL framework for distribution grid operation, and train it using samples of system states and control variables, as shown on the left side of Fig. 1. Once trained, the DRL agent is detached from the training setup and used as a decision maker (test setup in right side of Fig. 1), subject to feasibility checks that are presented later in this section.

In order to use DRL, first we need to define the problem as a MDP, where the distribution grid operation at any instance is represented by a state vector, and the transition to the next system state occurs only based on the current state, a set of operational actions, and a probabilistic transition function. Also, taking an action in a certain system state results in a reward that represents the action's value. The components of the MDP representation of grid operation are as follows:

- **System State:** We define the system state based on real-time loads ($\mathbf{p}^L$), stored energy of ES ($\mathbf{E}$), availability of distribution lines ($\mathbf{e}$), and energy price (λ). Additional factors that affects those parameters, such as time of the day or day of the week, or even weather data may be included in the state vector, and will result in more accurate operational decisions. The state vector is shown by $\mathbf{s} = [\mathbf{p}^L, \mathbf{E}, \mathbf{e}, \lambda, \mathcal{X}]$ where $\mathcal{X}$ includes additional available data.

- **Actions:** The operational control decisions are defined as the dispatch of DG units shown by $\mathbf{p}^g$ and charging and discharging power of ES units shown by $\mathbf{p}^e$, forming the action vector $\mathbf{u} = [\mathbf{p}^g, \mathbf{p}^e]$. Note that reactive power dispatch does not directly affect the operation cost and is not part of the action or state spaces. The reactive power dispatch will be obtained by running power flow, once the active power distribution is determined.

- **Reward Function:** The reward function is used in DRL framework to find efficient decisions and should

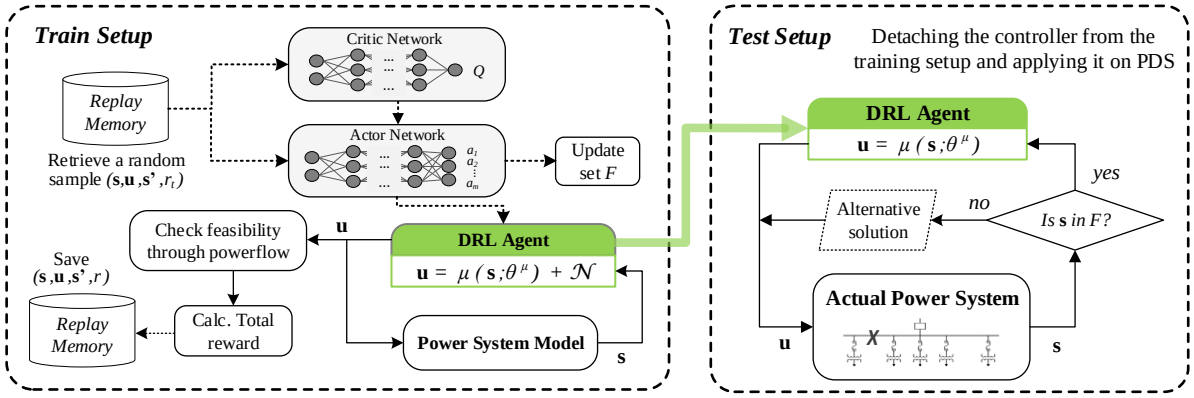

Figure 1. DRL training and testing setups with forming and applying feasibility set of the trained DRL agent.

be defined carefully. For distribution grid operation, this function should reward saving in the operation cost, and penalize undesirable conditions such as infeasibility. Also, adding a regularization term based on the L_m norm of the $\mathbf{u}$, helps with avoiding unnecessary large actions. The reward function in the proposed DRL model is represented by:

$$r(\mathbf{s},\mathbf{u}) = -C(\mathbf{s},\mathbf{u}) - P(v^{\text{dist}}, S^{\text{dist}}) - M\|\mathbf{u}\|_m, \quad (3)$$

where $C(.)$ and $P(.)$ are the cost and penalization functions and $v^{\text{dist}}, S^{\text{dist}}$ are total deviations of node voltages and line power flows from their limits, and are given by:

$$v^{\text{dist}} = \sum_{i\in\mathcal{I}} \max(\underline{v} - v_i, 0) + \max(v_i - \overline{v}, 0), \quad (4)$$

$$S^{\text{dist}} = \sum_{l\in\mathcal{L}} \max(S_l - \overline{S_l}, 0), \quad (5)$$

where v_i is the voltage of node $i \in \mathcal{I}$ and S_l is the apparent power flowing in line section $l \in \mathcal{L}$. To obtain deviation values, the following minimization problem is solved for each sample pair of states and actions, where v^{dist} and S^{dist} are linearized using auxiliary variables α, β, ν:

$$\min_{\alpha,\beta,\nu} \quad \sum_{i\in\mathcal{I}} \alpha_i + \beta_i + \sum_{l\in\mathcal{L}} \nu_l, \quad (6)$$

$$\text{s. t.} \quad \alpha_i \geq \underline{v} - v_i, \quad \forall i, \quad (7)$$

$$\beta_i \geq v_i - \overline{v}, \quad \forall i, \quad (8)$$

$$\nu_l \geq S_l - \overline{S_l}, \quad \forall l, \quad (9)$$

$$\alpha_i, \beta_i, \nu_l \geq 0, \quad \forall i, l, \quad (10)$$

Constraints (37)-(43) *in Appendix A*.

The optimization problem above is a form of branch power flow formulation, where state-action parameters are fixed, and strict voltage and line flow constraints (44) and (45) are replaced by relaxed constraints (7)-(9). Note that violation of direct limits of actions, such as generation limits of DER units is not penalized, as they can be adjusted into the feasible region.

- **Transition Matrix:** probability matrix $p(\mathbf{s}'|\mathbf{s},\mathbf{u})$, specifies the probability of transitioning between states given a certain action is taken. This matrix is governed by uncertainty sources such as load and solar power variations or occurrence of a fault, and is hard to obtain. DRL does not directly use the transition probability matrix but implicitly learns it through observing the transitions.

The DRL agent aims to find the operational decisions that leads to highest longer-term reward, and hence, should solve the following problem:

$$\max_{\mathbf{u}_t\in\mathbf{U}} \quad \mathbb{E}_{\sim p(\mathbf{s}_{t+1}|\mathbf{s}_t,\mathbf{u}_t)} \left[\sum_{t=t_0}^{\infty} \gamma^t [r(\mathbf{s}_t,\mathbf{u}_t)]\right], \quad (11)$$

where the expectation is over probability transition matrix $p(.)$, and $\gamma \in [0,1]$ is a discount factor that determines the significance of long-term versus immediate reward. The long-term reward is obtained recursively by the following Q-function that represents the action values in each system state:

$$Q(\mathbf{s},\mathbf{u}) = r(\mathbf{s},\mathbf{u}) + \gamma \mathbb{E}_{\sim p(\mathbf{s}'|\mathbf{s},\mathbf{u})} \left[\max_{\mathbf{u}'\in\mathbf{U}} Q(\mathbf{s}',\mathbf{u}')\right], \quad (12)$$

In the recursive formulation, values of s_t, u_t, s_{t+1} in each time t are replaced by s, u, s', respectively. In the discrete form of DRL, a decision is selected among multiple options by training a NN-based Q-network $Q(\mathbf{s},\mathbf{u};\theta)$, where θ is the weight vector, and then

selecting a decision as $\mathbf{u} = \arg\max_{\mathbf{u}'} Q(\mathbf{s}, \mathbf{u}'; \theta)$. In the continuous forms of DRL, such as DDPG method, separate actor and critic networks are trained in parallel, where an actor network generate continuous decisions as $\mathbf{u} = \mu(\mathbf{s}; \theta^{\mu})$ and the critic network estimates the long-term reward of the decision as $Q(\mathbf{s}, \mu(\mathbf{s}; \theta); \theta^{Q})$. Since the dispatch of generation units requires a continuous signal, we focus on the continuous form of DRL, and the training of actor and critic neural networks. The critic network is trained by minimizing the loss function (13):

$$L(\theta^{Q}) = \frac{1}{N} \sum_{i} \Big[Q(\mathbf{s}, \mu(\mathbf{s}; \theta^{\mu}); \theta^{Q}) - \Gamma_i \Big], \quad (13)$$

$$\Gamma_i = -r(\mathbf{s}, \mu(\mathbf{s}; \theta^{\mu})) + Q(\mathbf{s}', \mu(\mathbf{s}'; \theta^{\mu}); \theta^{Q}), \quad (14)$$

where $i = 1, ..., N$ are training samples and Γ_i defined in (14) is the long-term reward function obtained by the critic and actor networks trained so far, and $\mathbf{s}'$ is the next state of the system. The actor network is trained by updating its weights with the following gradient:

$$\nabla_{\theta^{\mu}} L(\theta^{\mu}) = \frac{1}{N} \sum_{i} \nabla_a Q(\mathbf{s}, \mu(\mathbf{s}; \theta^{\mu}); \theta^{Q}) . \nabla_{\theta^{\mu}} \mu(\mathbf{s}; \theta^{\mu}). \quad (15)$$

The training setup is schematically shown on the left side of Fig. 1. Note that stabilizing techniques such as *replay memory* and *target networks* are also used in training DRL agents, but we refrain from discussing them here and refer to e.g., [13] for detailed explanation. Once the DRL agent is trained, it makes decisions by a pre-trained actor network, that unlike the optimization problem, does not check system constraints for every decision. Although the actor network is trained to avoid infeasible decisions, and its actions are clipped within their direct upper and lower limits, the feasibility of solutions in all system states is not guaranteed. In the next section, we will define a feasible space around each feasible sample, and develop a method to discover the largest set of system states in which DRL produces a feasible and robust solution.

3. Discovering Feasible Space

Similar to any NN, a trained actor network in the DRL framework in Section 2 can be represented by its weight vectors $\mathbf{w}_k$ and bias vectors $\mathbf{b}_k$ of its structural layers, $k = 1, ..., K$, and a nonlinear activation function $\sigma(.)$ that follows every layer. Various NN structures as well as various activation functions such as ReLU, Leaky ReLU, Sigmoid exist and can be used for various deep learning applications. In this section, we find a feasible set of system states for which the DRL-trained agent produce feasible solutions, and develop our method based on DRL with feed-forward NN and the common ReLU activation function. The proposed approach can be extended to convolutional NNs and other well-known activation functions via certification process as in [18].

In a feed-forward NN, each layer completes the following affine transformation:

$$\hat{\mathbf{z}}_{k+1} = \mathbf{w}_{k+1} \mathbf{z}_k + \mathbf{b}_{k+1}, \ \ \forall k = 0, 1, ..., K, \quad (16)$$

$$\mathbf{z}_0 = \mathbf{s}, \ \ \hat{\mathbf{z}}_K = \hat{\mathbf{u}}, \quad (17)$$

and a ReLU activation function clips each neuron r in the layer's output into the positive half-space:

$$z_k^r = \max(\hat{z}_k^r, 0), \ \ \forall r, k. \quad (18)$$

Note that $\mathbf{z}_k$ and $\hat{\mathbf{z}}_k$ are output vectors of layer k before and after non-linear activation. The ReLU function can be represented by a set of linear equations with the help of an auxiliary binary variable $\mathbf{c}_k$ [15, 16]:

$$z_k^r \le \hat{z}_k^r - \hat{z}_k^{\min,r}(1 - c_k^r), \qquad \forall k, r, \quad (19)$$

$$z_k^r \ge \hat{z}_k^r, \qquad \forall k, r, \quad (20)$$

$$z_k^r \le \hat{z}_k^{\max,r} c_k^r, \qquad \forall k, r, \quad (21)$$

$$z_k^r \ge 0, \qquad \forall k, r, \quad (22)$$

$$c_k^r \in \{0, 1\}, \qquad \forall k, r. \quad (23)$$

The output of the NN is in the feasible space if the resulting vector $\mathbf{u}$ satisfies voltage constraints and flow limits of the power lines. We test the trained DRL agent on a set of randomly selected input samples and check if the output satisfies grid constraints. If solution to sample n is feasible, we call that a reference point $\mathbf{s}_n^{\text{ref}}$ and find the largest m-dimensonal feasible sphere that surrounds it by forming and solving the following maximization problem P2 for the input sample n, where R_n is the radius of the sphere:

$$\textbf{P2:} \ \max \ \ R_n \quad (24)$$

$$\text{s. t.} \ \ \|\mathbf{s} - \mathbf{s}_n^{\text{ref}}\|_m \le R_n, \quad (25)$$

$$(16), (17),$$

$$(19) - (23),$$

Constraint (37)-(45) *in Appendix A*.

Problem P2 is a quadratically constrained optimization problem that can be solved to optimality using commercial solvers. After finding R_n for each sample n, the feasible sphere is added to the total feasible set $\mathcal{F}$:

$$\mathcal{F} = \bigcup_{n=1}^{N} \{\mathbf{s} : \|\mathbf{s} - \mathbf{s}_n^{\text{ref}}\|_m \le R_n\}. \quad (26)$$

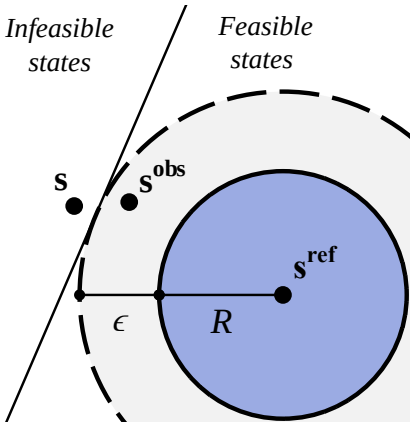

Figure 2. Robustness of feasibility spheres in 2D; spheres are formed with marginal radius ϵ to account for potential deviations in the observed system state.

If the total feasible set $\mathcal{F}$ is large enough to contain a large portion of the actual feasible space, it can be used by the operator to immediately weed out the infeasible system states from feasible ones. Although this method ensures the feasibility of DRL solutions, it does not guarantee robustness against deviations in the observed system states. For example, assume that an observed state $\mathbf{s}^{\text{obs}}$ deviates from the actual state, i.e., $\|\mathbf{s} - \mathbf{s}^{\text{obs}}\|_m > 0$, due to noise, perturbations or a data injection attack, and the feasibility set contains $\mathbf{s}^{\text{obs}}$ but not $\mathbf{s}$. In this case, the operator can verify that the observed state is in the feasible space, while the actual state results in an infeasible solution when using the DRL agent. To improve the robustness, we find feasibility spheres with radius $R + \epsilon$ to account for ϵ deviation of $\mathbf{s}$ and create feasibility sets for certain maximum deviations, as illustrated in Fig. 2. Accordingly, (25) is re-written as:

$$\|\mathbf{s} - \mathbf{s}_n^{\text{ref}}\|_m \leq R_n + \epsilon, \tag{27}$$

and the ϵ-robust feasibility sets are given by:

$$\mathcal{F}^{\epsilon} = \bigcup_{n=1}^{N} \{\mathbf{s} : \|\mathbf{s} - \mathbf{s}_n^{\text{ref}}\|_m \leq R_n + \epsilon\}. \tag{28}$$

It is possible to create multiple $\mathcal{F}^{\epsilon}$ sets for different ϵ values, and apply the appropriate one during the operation, based on the extent of threats that the system is exposed to at any time.

4. Numerical Study

The proposed method is demonstrated on three test distribution networks, namely 13 bus [19], 33 bus [20], and 123 bus [19], and its capability is analyzed in discovering the feasible space of a DRL agent that is trained to operate the test networks. The original test systems are modified by adding multiple DG and ES units. The specifications and locations of the added units in each test systems are shown in Table 1. Note that added photovoltaic (PV) units are treated as units with non-controllable active power and controllable reactive power. Figure 3 shows the modified 123-bus network with the added DER. The total storage capacity of ES

Table 1. Specifications of added DER units to the test systems.

	13 bus	33 bus	123 bus	
	Bus no.	**Bus no.**	**Bus no.**	**Max/Min P**
DG	611	22	20,86,104	250/0
ES	675	14,33	26,49,67,94,112	600/-600
PV	-	18	39,59	150/0

units is 6kWh. Also, since the original 123-bus test systems do not specify line capacity data, we assume the typical value of 150A in each of the three phases for all distribution lines, with the exception of substation outgoing lines, which are capped by 300A. Hourly load of zone CAPITL of the energy market & operational data of NYISO for year 2017 [21] is scaled down and used in the system. The energy price data (λ) of the same zone is also used in the simulations. Also, the global horizontal irradiation profile in NYC in 2017 [22] is used as the generation profile of PV units, and the profile is normalized to the PV capacity of each network to generate a year-long hourly data. In this study, the operation cost function is defined as:

$$C(\mathbf{s}, \mathbf{u}) = \lambda p^{grid} + \lambda^g p^g,$$

where p^{grid} is the total active power from the substation, and λ^g is fixed at 5 cent/kW. Also, by ignoring the cost function, we have:

$$p^{grid} = \sum_{i \in \mathcal{I}} p_i^L - \sum_{i \in \mathcal{I}(\mathcal{G})} p_i^g$$

.

4.1. Infeasibility in DRL-based operation

The solution given by the DRL agent for a 24-hour operation of the 123-bus network is shown in Fig. 4, where system states, including loading percentage, energy price, and energy level of ES units are shown on the top graph, and the dispatch signal for ES2, ES3, and DG3 are shown below it. In the bottom graph in Fig. 4, the maximum and minimum voltages throughout

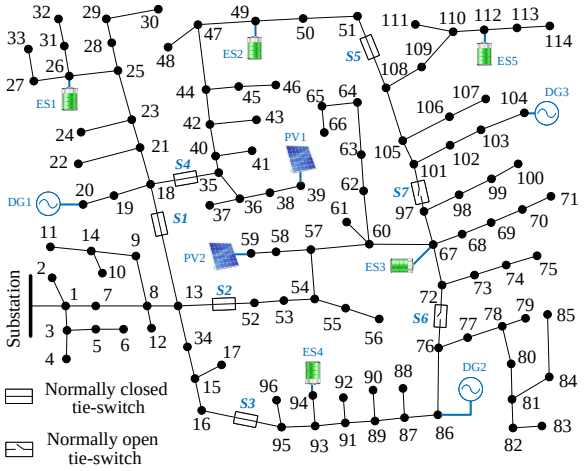

Figure 3. The modified 123-bus test system with added DER units.

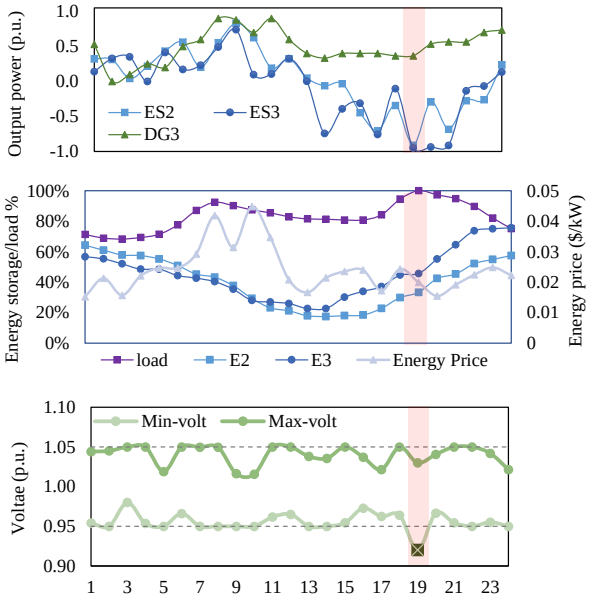

Figure 4. Results of DRL-based operation of the modified 123-bus network in 24 hours. The graphs show system states (top), operation decisions (middle), and min/max voltage in the grid (bottom).

the network are shown. Note that in most hours, the minimum and maximum voltages are very close to limits, which shows the efficient training of the actor network. However, as highlighted in the figure, at hour 19, the minimum voltage drops below the minimum limit, which is an infeasible condition. Forming a feasibility set and applying it in the operation, allows for identifying states that are not proven to have a feasible solution, and bypassing the DRL agent in those states.

4.2. Discovery of Feasible Spheres

The feasible spheres around three sample feasible points are shown in Fig. 5 for a instance of the 33-bus test system, where the three dimensions represent the loading condition, energy reserve of ES1, and output percentage of PV1. For displaying the feasible space in three dimensions, line availabilities are fixed and the same loading condition is assumed for all load. In Fig. 5, the red surface separates the actual feasible range of DRL solutions (above the red surface) from the rest and is found using Monte Carlo Method. In this method, numerous random values are generated within the action domain and their feasibility or infeasibility is determined by running a power flow for each point. Then a polynomial support vector machine is applied on these points to find the separation half-space between feasible and infeasible sections. This method is time consuming and computationally expensive, and hence cannot be used in real-time operation. However, it is used in Figs. 5 and 6 to showcase the ability of feasibility ellipsoids in finding feasible sections.

Figure 5 shows the portion of feasible space detected by first three random sampling and forming largest feasible spheres around them. The spheres are formed for ϵ values of 0 and 0.05 p.u. robustness.

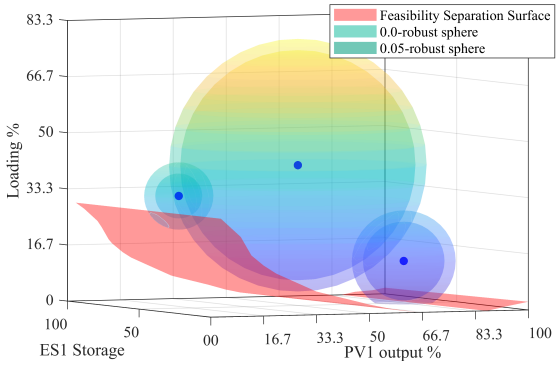

Figure 5. Forming largest feasible spheres around sample points to discover the feasible space.

In order to find the required number of sample points for an effective discovery of feasible space, problem P2 is solved for 1000 samples on all of the three test networks, and the portion of the feasible space discovered by this method is calculated. These 1000 samples include infeasible points as well, which we discard without applying Problem P2. Figure 6 shows the trend of discovering the feasible space in all cases and indicates that more than 90% of the feasible space is discovered after 1000 sampling in all three networks.

This would provide a useful tool for system operators who use DRL agents in their decision making, to immediately evaluate the feasibility of DRL solution.

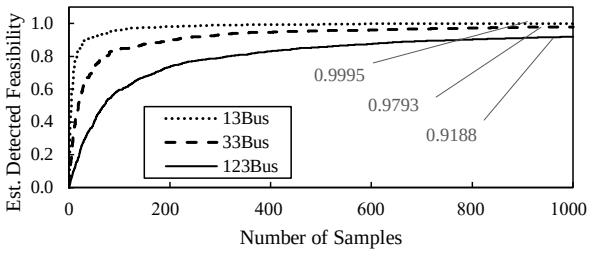

Figure 6. Portion of the feasible space discovered by forming feasibility spheres around input samples.

4.3. Forming ϵ-Robust Feasibility Sets

The ϵ-robust feasibility sets are formed for each network for different values of ϵ, expressed in p.u. of system states, as discussed in Section 3. The number of mistakes that different sets make in detecting infeasible system states are shown in Fig. 7 for 1000 sample states of the 33-bus network. Evidently, higher robustness results in less vulnerability to perturbation, at the cost of missing more feasible points. Once the deviation ϵ exceeds the robustness of the feasibility set, the number of missed infeasibilities increase at an exponential rate.

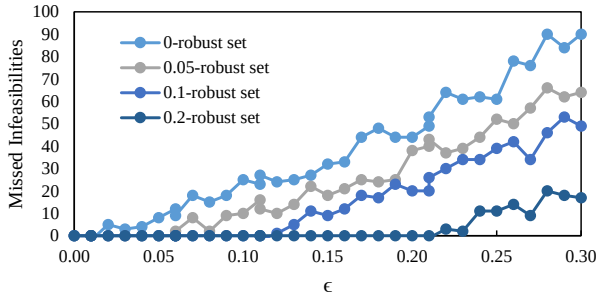

Figure 7. Number of mistakes in detecting infeasible system states in 33-bus network, as the deviation ϵ increases (total sample size is 1000).

4.4. Infeasibility Penalization Methods

In order to reduce the number of infeasible solutions by the DRL agent, a penalization mechanism must be added in the training setup. Usually, that is incorporated in the reward function (3), where actions are penalized for violation of voltage or line overflow constraints by a penalization function $P(.)$. The form of function P directly affects the feasiblity and optimality of the trained agent. A light penalization cannot properly avoid infeasibilities, while a strong penalization results in solutions that are far away from border points, where optimal solutions usually reside. We train DRL agents with four penalization function:

$$\text{logarithmic:} \quad \log(v^{\text{dist}} + S^{\text{dist}}), \tag{29}$$

$$\text{linear:} \quad A(v^{\text{dist}} + S^{\text{dist}}) + B, \tag{30}$$

$$\text{quadratic:} \quad A(v^{\text{dist}} + S^{\text{dist}})^2, \tag{31}$$

$$\text{cubic:} \quad A(v^{\text{dist}} + S^{\text{dist}})^3. \tag{32}$$

Table 2 shows operation cost and infeasibility of decisions made by the DRL agents that are trained by different penalization methods with $A = 1, B = 0$. Due to stochastic nature of DRL, for each test system, 20 agents are trained by each penalization method, and the average, maximum and standard deviation of operation costs and infeasiblities are reported. As expected, higher degree of penalization of constraint violations results in less under/over voltage and line overflow. Cubic and quadratic penalization reduce the average infeasibility, however, they result in higher operation costs, as they tend to avoid border points with potentially higher optimality. In essence, the actions become conservative. In the trade-off between optimality and feasiblity, the quadratic penalization performs better, as its operation cost is only slightly higher than the linear case, but its total infeasibility is considerably smaller.

5. Conclusion

This paper develops a model for verification of DRL solutions for power distribution system operation, by forming a sufficiently large feasible space in the system states, for which the DRL solution meets the system constraints. The feasible space is discovered by finding largest spheres around sample operating points, and then aggregating those spheres. Numerical studies show that even for the relatively large distribution systems, the feasible space may be effectively discovered by sampling around 1000 feasible points. In smaller systems, higher detectability is achieved by smaller samples. The obtained feasible set is used in an operation setup, where DRL solutions are checked with the feasible set before applying them in the operation. Further, the impact of infeasibility penalization function on the percentage of impossible solutions is studied, and the quadratic function proved more effective, both in optimaility and feasibility, than logarithmic, linear, or cubic functions.

Table 2. Operation cost and infeasibility of DRL agent trained with different penalization methods.

	Operation Cost								
	13bus			33bus			123bus		
Penalization method	**Ave.**	**Max.**	**sdev.**	**Ave.**	**Max.**	**sdev.**	**Ave.**	**Max.**	**sdev.**
Cubic	13471	19221	± 2954	32118	40592	± 6608	116037	205307	± 25310
Quadratic	11729	17852	± 2414	28517	39719	± 5709	111943	175290	± 21085
Linear	11577	16710	± 2489	26486	35044	± 6108	113241	175344	± 22510
Logarithmic	12952	18628	± 2745	31153	39069	± 7178	115357	196340	± 23012
	Line Flow Infeasibility (total overcurrent in 24 hours in p.u.)								
	13bus			33bus			123bus		
Penalization method	**Ave.**	**Max.**	**sdev.**	**Ave.**	**Max.**	**sdev.**	**Ave.**	**Max.**	**sdev.**
Cubic	0.17	0.86	± 0.24	0.37	0.97	± 0.33	1.15	4.08	± 1.01
Quadratic	0.35	1.11	± 0.36	0.76	1.60	± 0.54	5.36	9.64	± 3.59
Linear	1.25	5.50	± 1.48	3.52	7.72	± 2.39	15.22	22.71	± 6.56
Logarithmic	1.76	5.77	± 1.98	7.37	9.21	± 3.08	19.92	29.72	± 9.34
	Voltage Infeasibility (total over/under voltage in 24 hours in p.u.)								
	13bus			33bus			123bus		
Penalization method	**Ave.**	**Max.**	**sdev.**	**Ave.**	**Max.**	**sdev.**	**Ave.**	**Max.**	**sdev.**
Cubic	1.12	1.44	± 0.13	2.56	3.74	± 0.44	7.64	9.20	± 1.16
Quadratic	1.42	2.08	± 0.23	3.16	4.34	± 0.51	8.04	9.92	± 1.44
Linear	1.62	2.01	± 0.38	3.78	4.78	± 0.60	9.48	11.84	± 1.43
Logarithmic	1.74	2.98	± 0.49	3.76	5.04	± 0.77	11.87	14.04	± 2.24

Appendix A Distribution Grid Constraints

To check the feasibility of DRL solutions for operation of DG and ES units in the distribution grid, we use the quadratically constrained (QC) formulation of branch power flow to check if network constraints are satisfied by a certain decision. The original branch flow equations are as follows [20]:

$$\sum_{j'|(i,j')\in\mathcal{L}} p_{ij'} = p_{ji} - r_{ij}\frac{p_{ji}^2+q_{ji}^2}{V_i^2} - p_i^L,\ \forall i,j, \quad (33)$$

$$\sum_{j'|(i,j')\in\mathcal{L}} q_{ij'} = q_{ji} - x_{ij}\frac{p_{ji}^2+q_{ji}^2}{V_i^2} - q_i^L,\ \forall i,j, \quad (34)$$

$$V_i^2 = V_j^2 - 2(r_{ij}p_{ij} + x_{ij}q_{ij}) + (r_{ij}^2+x_{ij}^2)\frac{p_{ji}^2+q_{ji}^2}{V_i^2},\ \forall i,j, \quad (35)$$

which govern active and reactive power balance on nodes, and voltage drop on lines, respectively. In (33)-(35), $\mathbf{p}, \mathbf{q}$ are vectors of active and reactive power flowing in lines, $\mathbf{V}$ is the node voltage vector, $\mathbf{p}^L, \mathbf{q}^L$ are active and reactive loads, and $\mathbf{r}, \mathbf{x}$ are resistance and reactance of the line sections. To achieve the quadratic approximation of (33)-(35), we set:

$$\frac{p_{ji}^2+q_{ji}^2}{V_i^2} \approx p_{ji}^2+q_{ji}^2,\ \forall i,j, \quad (36)$$

and ignore the degree four term in (35). Adding DER units and load shedding option to the formulations and setting $v_i = V_i^2,\ \forall i$, the approximated equations are presented as follows:

$$\sum_{j'|(i,j')\in\mathcal{L}} p_{ij'} \le p_{ji} - r_{ij}(p_{ji}^2+q_{ji}^2) - \eta_i p_i^L + p_i^g + p_i^e, \quad (37)$$

$$\sum_{j'|(i,j')\in\mathcal{L}} q_{ij'} \le q_{ji} - x_{ij}(p_{ji}^2+q_{ji}^2) - \eta_i q_i^L + q_i^g + q_i^e, \quad (38)$$

$$v_j \le v_i - 2(r_{ij}p_{ij} + x_{ij}q_{ij}) + M(1-e_{ij}), \quad (39)$$

$$v_j \ge v_i - 2(r_{ij}p_{ij} + x_{ij}q_{ij}) - M(1-e_{ij}), \quad (40)$$

$$E_i = E_i^{-1} - p_i^e, \quad \forall i \in \mathcal{E}(I), \quad (41)$$

$$\{\underline{p}_i^g, \underline{p}_i^e, \underline{q}_i^g, \underline{q}_i^e\} \le \{p_i^g, p_i^e, q_i^g, q_i^e\} \le \{\overline{p}_i^g, \overline{p}_i^e, \overline{q}_i^g, \overline{q}_i^e\}, \quad (42)$$

$$0 \leq \eta_i \leq 1, \tag{43}$$

$$p_{ij}^2 + q_{ij}^2 \leq \overline{S}_{ij}^2 . e_{ij}, \tag{44}$$

$$\underline{v} \leq v_i \leq \overline{v}. \tag{45}$$

In (37), (38) η is the load shedding factor whose limits are given by (43), and $\mathbf{p}^g, \mathbf{p}^e$ are output vector of DG and ES units. The voltage drop on each line is formulated in (39) and (40), where $\mathbf{e}$ is the line availability vector and M is a large number. Evolution of energy level of ES units are formulated in (41), where $\mathbf{E}, \mathbf{E}^{-1}$ are current and previous energy vectors of ES units located on a subset of buses $\mathcal{E}(I)$ and a perfect charging efficiency is assumed. Equation (44) limits the power flow in each line section, where $\mathbf{S}$ is the apparent flowing power. Finally, the voltage vector $\mathbf{v}$ is constrained by upper and lower limits in (45). Note that equations (37)-(45) are true for all of their indices, unless otherwise specified.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[3] N. Mazyavkina, S. Sviridov, S. Ivanov, and E. Burnaev, "Reinforcement learning for combinatorial optimization: A survey," *Computers & Operations Research*, p. 105400, 2021.

[4] T. Barrett, W. Clements, J. Foerster, and A. Lvovsky, "Exploratory combinatorial optimization with reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 3243–3250, 2020.

[5] Q. Huang, R. Huang, W. Hao, J. Tan, R. Fan, and Z. Huang, "Adaptive power system emergency control using deep reinforcement learning," *IEEE Transactions on Smart Grid*, 2019.

[6] Y. Du and F. Li, "Intelligent multi-microgrid energy management based on deep neural network and model-free reinforcement learning," *IEEE Transactions on Smart Grid*, 2019.

[7] Q. Yang, G. Wang, A. Sadeghi, G. B. Giannakis, and J. Sun, "Two-timescale voltage control in distribution grids using deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2313–2323, 2019.

[8] M. M. Hosseini and M. Parvania, "Artificial intelligence for resilience enhancement of power distribution systems," *The Electricity Journal*, vol. 34, no. 1, p. 106880, 2021.

[9] M. Majidi, D. Muthirayan, M. Parvania, and P. P. Khargonekar, "Dynamic matching markets in power grid: Concepts and solution using deep reinforcement learning," *arXiv preprint arXiv:2104.05654*, 2021.

[10] G. K. Venayagamoorthy, R. K. Sharma, P. K. Gautam, and A. Ahmadi, "Dynamic energy management system for a smart microgrid," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 8, pp. 1643–1656, 2016.

[11] H. Hua, Y. Qin, C. Hao, and J. Cao, "Optimal energy management strategies for energy internet via deep reinforcement learning approach," *Applied Energy*, vol. 239, pp. 598–609, 2019.

[12] B. V. Mbuwir, F. Ruelens, F. Spiessens, and G. Deconinck, "Battery energy management in a microgrid using batch reinforcement learning," *Energies*, vol. 10, no. 11, p. 1846, 2017.

[13] M. M. Hosseini and M. Parvania, "Resilient operation of distribution grids using deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, 2021 (in press).

[14] R. Bunel, I. Turkaslan, P. H. Torr, P. Kohli, and M. P. Kumar, "A unified view of piecewise linear neural network verification," *arXiv preprint arXiv:1711.00455*, 2017.

[15] V. Tjeng, K. Y. Xiao, and R. Tedrake, "Evaluating robustness of neural networks with mixed integer programming," in *International Conference on Learning Representations*, 2018.

[16] A. Venzke and S. Chatzivasileiadis, "Verification of neural network behaviour: Formal guarantees for power system applications," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 383–397, 2020.

[17] A. Venzke, G. Qu, S. Low, and S. Chatzivasileiadis, "Learning optimal power flow: Worst-case guarantees for neural networks," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–7, IEEE, 2020.

[18] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," *arXiv preprint arXiv:1811.00866*, 2018.

[19] W. H. Kersting, "Radial distribution test feeders," *IEEE Transactions on Power Systems*, vol. 6, no. 3, pp. 975–985, 1991.

[20] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Power Engineering Review*, vol. 9, no. 4, pp. 101–102, 1989.

[21] "Energy market & operational data." `http://https://www.nyiso.com/energy-market-operational-data`. Accessed: 2021-06-12.

[22] M. Sengupta, Y. Xie, A. Lopez, A. Habte, G. Maclaurin, and J. Shelby, "The national solar radiation data base (nsrdb)," *Renewable and Sustainable Energy Reviews*, vol. 89, pp. 51–60, 2018.